A.0004 循环控制结构

Welcome back! 这次拖了好久,主要因为个人杂事和懒惰,二是因为反复构思这最重要最棘手的一节应该怎么讲,怎样的练习题才能吸引读者.

这次我们来讲编程中最重要的知识点之一,循环[Loop].什么是循环呢?感性地想,就是程序中需要反复重复一项类似的操作.例如,对一个数据表的每一行都做同样的处理操作.又例如,黑客重复尝试某帐号的密码.

循环的语法写法有很多很多种,然而它们都是等效的.不同的写法只是为了让程序更简洁一些.对于一个小的数据处理程序来讲,采取那种写法几乎无任何差别.于是我决定选取其中的两个来讲,剩下的看需求可能在之后的章节提及.下面我们就讲讲其中最类似 C 语言的循环控制语法,for 语句和 while 语句.

```
while(condition)
{
    code
}
```

这就是 while 语句的写法.执行次序是这样的:先判断 con,如果成立,则执行 code 部分代码;之后再判断 con,成立则再执行 code 部分的代码.如果哪次判断 con 不成立,则直接跳到{}之后,执行{}之后的代码.

我们来用 while 举个例子.传说高斯小时候速算 1 到 100 的和,现在我们用暴力法来计算一下.

```
$a=1;     # number added each time
$sum=0;   # sum
while($a<=100)
{
    $sum=$sum+$a;
    $a=$a+1;
}
print ("$sum\n");
```

用$a 来放每次的加数,$sum 用来放和.初值分别是 1 和 0.仔细读读想想,应该不是很难理解.

下面是较复杂一些的 for 循环:

```
for(op1;condition;op2)
{
    code
}
```

for 循环的执行次序是这样的:先执行 op1.然后判断 con,如果成立执行 code,再执行 op2;然后再判断 con,如果成立执行 code,再执行 op2.如果哪次判断 con 时不成立,则跳转到{}后面,执行{}后面的代码.

for 为什么这么复杂呢?因为它每个部分都是有含义的.

我们发现 op1 只执行一次.op1 一般只有一句或两句,是用来给这个循环做初始化的,尤其常用于给循环变量做初始化.

我们又发现,code+op2 总是在一起执行(或者 con 不成立导致不执行)的,那么为什么要把二者分开呢?因为我们一般把循环变量的变化写在 op2 的位置上.

这是什么意思呢?我们把上述 while 程序改写成 for 程序,以此为例讲解.

```
$sum=0;   # sum
for($a=1;$a<=100;$a++)
{
    $sum=$sum+$a;
}
print ("$sum\n");
```

注意看这段代码.$a 相当于一个循环变量,我们在 op1 的位置上给它初始化.在 code 位置做加法,而在 op2 的位置上让$a 自加 1.结合理解一下为什么 for 循环要分成这么多段.其实把 op2 写到 code 里面去,也是等效的,分开只是为了理清思路.

++是自加 1 操作符,--是自减 1 操作符,循环中常用.$x++等效于$x=$x+1,只是为了方便.减法同.

如果 op1 或者 op2 处要写不止一句代码,那么多句之间用逗号,隔开.例如,for($sum=0,$a=1;$a<=100;$a++)

如果 for 循环的这四个部分中的某个部分不需要代码,留白即可.但是注意 for(;;)这个括号内的两个分号不能省略.

提醒一下,注意仔细观察一下方框中对 for 语法的描述.op1,op2,code 三个部分是可有可无的,但是哪里该有分号和括号,哪里没有分号,一定要搞清楚,千万不要写错了.()中的两个分号必须有.()后面没有分号.{}不要忘记.{}后面是没有分号的.

```
for(op1;condition;op2)
{
    code
}
```

上面讲的知识点密度比较大.现在我们来举几个例子吧.由简单的开始.

现在我们来计算一下[1,100]中所有 3 的倍数的和.

```
$a=1;
$s=0;
for(;$a<=100;$a++)
{
    if ($a%3==0) {$s=$s+$a;}
}
print ("$s\n");
```

读一遍,应该能懂的吧.上述代码也可以这样写:

```
$s=0;
for($a=3;$a<=100;$a=$a+3)
{
```

```
    $s=$s+$a;
}
print ("$s\n");
```

这个就是从 3 开始,步长为 3,一直累加到 100.这次循环变量不再是每次加 1 了.

下面我们尝试一下一个复杂一些的,计算斐波纳妾数列的前 20 项.这个数列以 1, 1, 2, 3 开头,每一项都是前两项的和.

通过观察,我们知道前两项是固定值 1,从第三项开始,都套用前两项的和.循环从第 3 项做到第 20 项.先写下这一行

```
for($i=3;$i<=20;$i++)
{}
```

然后在其前面先把前两项输出出来,然后设计用三个变量来分别表示前两项和现在要计算的这一项.

```
print ("1 1 ");
$x=1;
$y=1;          # 假设 x<=y
for($i=3;$i<=20;$i++)
{
    $z=$x+$y; #这次循环计算的这一项是前两项的和
    print("$z ");
    $x=$y;
    $y=$z;
}
```

其中,$x=$y;$y=$z;这两句是为下次循环做准备.意义是,对于下次循环来说,"前两项"其实就是这次循环中的 y 和 z.能理解么?

练习题.这次先以数学题为主练练手.

1. 求数列前 50 项的和 1+3+5+7+…
2. 求数列前 50 项的和 1+1/3+1/5+1/7+…
3. 求数列前 50 项的和 1-1/3+1/5-1/7+…
4. 输出[1,500]之间的全部完全平方数(如果一个整数 z 的平方根仍然是一个整数,那么这个整数 z 就叫做完全平方数.例如 1,4,9,16,25 等)

```
2500
2.93777484847491
0.780398663147753
1 4 9 16 25 36 49 64 81 100 121 144 169 196 225 256 289 324 361 400
441 484
```

5．从键盘读入一个整数(>=0).写一个程序来判断这个数是不是素数,如果是,输出 yes,不是 no
6．利用上面写过的程序,求[1,500]内所有素数的和． 21536
7．